What is claimed is:

1. A method for performing a modular multiplication on data processing means between a multiplicand and a multiplier consisting of a plurality of digits, using a modulus, the modular multiplication being part of a modular exponentiation within the framework of a cryptographic application, and the multiplicand, the multiplier and the modulus being variables of the cryptographic application, the method comprising:

determining l multiplication shift values by means of a multiplication-lookahead method while taking into account l blocks of consecutive digits of the multiplier, l being larger or smaller than 2;

determining l reduction shift values by means of a reduction-lookahead method for the l blocks of digits of the multiplier;

applying the l multiplication shift values and the l reduction shift values to an intermediate result from a previous iteration step, to the modulus or to a value derived from the modulus, and to the multiplicand so as to obtain 2l+1 operands; and

combining the operands to obtain an updated intermediate result for an iteration step following the previous iteration step, an iteration being continued for such time until all digits of the multiplier have been processed, wherein the updated intermediate result, once all digits of the multiplier have been processed, is a result of the modular exponentiation within the framework of the cryptographic application.

2. The method as claimed in claim 1, wherein

the step of determining l multiplication shift values further includes determining l multiplication-lookahead parameters;

the step of determining l reduction shift values further includes determining l reduction-lookahead parameters; and

the step of applying includes using the l multiplication-lookahead parameters and the l reduction-lookahead parameters to obtain the 21+1 operands.

3. The method as claimed in claim 1, wherein the step of determining the l reduction shift values includes performing the following substeps:

determining an auxiliary shift value from an intermediate result and from a modulus value for a preceding iteration step;

forming a difference from the multiplication shift value for a first number of digits of the multiplier, and the auxiliary shift value so as to obtain a reduction shift value.

4. The method as claimed in claim 3, wherein the step of determining l reduction shift values further includes the following substeps:

calculating an auxiliary intermediate result using the intermediate result for the preceding iteration step, calculating the first multiplication shift value, the modulus and the reduction shift value, however without taking into account the multiplicand;

calculating an auxiliary modulus by shifting the modulus or the value derived from the modulus by a number of digits equaling the reduction shift value;

calculating a further auxiliary shift value from the auxiliary intermediate result and the auxiliary modulus; and

forming a difference from the second multiplication shift value and the second auxiliary shift value so as to obtain the second reduction shift value.

5. The method as claimed in claim 1,

wherein prior to the step of determining 1 multiplication shift values, and prior to the step of determining 1 reduction shift values, the following step is performed:

transforming the modulus to a transformed modulus larger than the modulus, a predetermined fraction (2/3) of the transformed modulus having a more significant digit having a first predetermined value, which digit is followed by a less significant digit having a second predetermined value;

and wherein the following step is performed once all digits of the multiplier have been processed:

transforming the updated intermediate result back by modular reduction of the updated intermediate result using the modulus,

the steps of determining, of applying and of combining being performed on the basis of the transformed modulus.

6. The method as claimed in claim 5, wherein the step of determining 1 reduction shift values includes a substep of determining a multiplication intermediate result and a reduction shift value, the reduction shift value being calculated using a determination of the number of digits between the more significant digit having the first predetermined value of the transformed modulus, and the

most significant digit of the intermediate result having the first predetermined value.

7. The method as claimed in claim 5, wherein the predetermined fraction of the modulus is 2/3.

8. The method as claimed in claim 5, wherein the most significant bit of the transformed modulus is a sign bit, and wherein a more significant portion of the predetermined fraction of the modulus is as follows:

01000 xx ... xx,

wherein the bits designated by xx may have any values.

9. The method as claimed in claim 8, wherein the more significant portion of the transformed modulus is as follows:

01100 ... 00.

10. The method as claimed in claim 1,

wherein the modulus is an integer, or wherein the modulus is a polynomial of a variable.

11. An apparatus for performing a modular multiplication on data processing means between a multiplicand and a multiplier consisting of a plurality of digits, using a modulus, the modular multiplication being part of a modular exponentiation within the framework of a cryptographic application, and the multiplicand, the multiplier and the modulus being variables of the cryptographic application, the apparatus comprising:

means for determining l multiplication shift values by means of a multiplication-lookahead method while taking

into account l blocks of consecutive digits of the multiplier, l being larger or smaller than 2;

means for determining l reduction shift values by means of a reduction-lookahead method for the l blocks of digits of the multiplier;

means for applying the l multiplication shift values and the l reduction shift values to an intermediate result from a previous iteration step, to the modulus or to a value derived from the modulus, and to the multiplicand so as to obtain 2l+1 operands; and

means for combining the operands to obtain an updated intermediate result for an iteration step following the previous iteration step, an iteration being continued for such time until all digits of the multiplier have been processed, wherein the updated intermediate result, once all digits of the multiplier have been processed, is a result of the modular exponentiation within the framework of the cryptographic application.